



TUGAS AKHIR - TE145561

**PERANCANGAN *HUMAN MACHINE INTERFACE*
SIEMENS MINI CNC MTU 200 UNTUK KEPERLUAN
PENGAWASAN DAN *MAINTENANCE***

Arinto Eka Prasetyo R.
NRP. 10311500000014

Pembimbing
Slamet Budiprayitno, S.T., M.T .
Moh. Abdul Hady, S.T., M.T.
Affan Naziihan Nuufa

DEPARTEMEN TEKNIK ELEKTRO OTOMASI
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018



FINAL PROJECT - TE 145561

***BUILDING HUMAN MACHINE INTERFACE SIEMENS
MINI CNC MTU 200 FOR CONDUCTING SUPERVISION
AND MAINTENANCE***

Arinto Eka Prasetyo R.
NOR. 10311500000014

Supervisor
Slamet Budiprayitno, S.T., M.T .
Moh. Abdul Hady, S.T., M.T.
Affan Naziihan Nuufa

***ELECTRICAL AND AUTOMATION ENGINEERING DEPARTMENT
Vocational Faculty
Institut Teknologi Sepuluh Nopember
Surabaya 2018***

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan dari Tugas Akhir saya dengan judul:

“Perancangan *Human Machine Interface* Siemens Mini CNC MTU 200 untuk Keperluan Pengawasan dan *Maintenance*”

adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap di dalam daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai dengan peraturan yang berlaku.

Surabaya, Juli 2018



Arinto Eka Prasetyo R.
NRP. 10311500000014

-----Halaman ini sengaja dikosongkan-----

PERANCANGAN *HUMAN MACHINE INTERFACE* SIEMENS MINI CNC MTU 200 UNTUK KEPERLUAN PENGAWASAN DAN *MAINTENANCE*

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Memperoleh Gelar Ahli Madya Teknik**

Pada

**Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember**

Pembimbing I

Pembimbing II



Slamet Budi Prayitno, S.T., M.T. DEPARTEMEN **Moh. Abdul Hady, S.T., M.T.**
TEKNIK ELEKTRO OTOMASI
NIP. 196510141990021001 **NIP. 1987041320150410002**

**SURABAYA
JULI, 2018**

-----Halaman ini sengaja dikosongkan-----

**PERANCANGAN *HUMAN MACHINE INTERFACE* SIEMENS
MINI CNC MTU 200 UNTUK KEPERLUAN PENGAWASAN
DAN *MAINTENANCE***

Nama : Arinto Eka Prasetyo R.
NRP : 10311500000014
Pembimbing I : Slamet BudiPrayitno, S.T., M.T.
Pembimbing II : Moh.Abdul Hady, S.T., M.T.
Pembimbing III : Affan Naziihan Nuufa, Amd

ABSTRAK

CNC dibuat dapat membantu meringankan biaya pengeluaran tanpa harus menyewa jasa untuk pembuatan barang. PT CNC Controller Indonesia Membuat mesin Mini CNC MTU200 untuk sarana belajar siswa dan mahasiswa untuk mengetahui cara kerja mesin CNC. Mesin Mini CNC digunakan sekolah dan institusi untuk pembelajaran tersebut

Oleh karena itu dibutuhkan suatu sistem yang digunakan untuk mengawasi komponen dari Mini CNC. Pengawasan ini diharapkan dapat membantu bagian *maintenance* untuk mendeteksi bagian mana yang sedang bermasalah dengan menggunakan HMI untuk settingnya dan memprogram PLC untuk mendeteksi error pada mesin Mini CNC ini. Setting HMI ini ditambahkan parameter tambahan untuk dapat dibandingkan data komponen dan batasan yang ada untuk tiap komponen. Pada sistem ini komponen yang akan dijadikan objek pada pengawasan ini adalah motor servo karena menurut perusahaan bagian ini paling banyak dan paling penting pada mesin CNC.

Dalam sistem ini menggunakan Sinumerik808D sebagai HMI dan PLC sebagai *controller*. Penggunaan alat ini bisa mengurangi kesalahan pada mesin Mini CNC 200 sebanyak 98 persen.

Kata Kunci : HMI, CNC, Maintenance

-----Halaman ini sengaja dikosongkan-----

***BUILDING HUMAN MACHINE INTERFACE SIEMENS MINI
CNC MTU 200 FOR CONDUCTING SUPERVISION AND
MAINTENANCE***

Name : Arinto Eka Prasetyo R.
NOR : 10311500000014
Supervisor I : Slamet BudiPrayitno, S.T., M.T .
Supervisor II : Moh.Abdul Hady, S.T., M.T.
Supervisor III : Affan Naziihan Nuufa, Amd

ABSTRACT

CNC made can help ease the cost of spending without having to hire services for the manufacture of goods. PT CNC Controller Indonesia Make CNU MTC200 Mini Machine for student and student learning tool to know how CNC machine work. Mini CNC machines are used schools and institutions for such learning

Therefore, a system used to monitor the components of Mini CNC is required. This supervision is expected to assist the maintenance department to detect which part is in trouble by using HMI for its settings and program the PLC to detect errors on this Mini CNC machine. This HMI setting added additional parameters to be able to compare the component data and restrictions that exist for each component. In this system the component that will be the object of this supervision is servo motor because according to the company this part most and most important on CNC machine.

In this system use Sinumerik808D as HMI and PLC as controller. This tool can be used on CNC machine because it can show parameter on CNC machine

Keywords : HMI, CNC, Maintenance

-----Halaman ini sengaja dikosongkan-----

KATA PENGANTAR

Alhamdulillah kami panjatkan kepada Allah Subhanahu Wa Ta'ala, atas limpahan rahmat dan kemudahan dariNya, hingga kami dapat menyelesaikan tugas akhir ini dengan baik, begitu pula dengan pembuatan buku tugas akhir ini.

Tugas akhir ini dilakukan untuk memenuhi beban satuan kredit semester (SKS) yang harus ditempuh sebagai persyaratan akademis di Departemen Teknik Elektro Otomasi Fakultas Vokasi Institut Teknologi Sepuluh Nopember Surabaya untuk menyelesaikan program pendidikan Diploma di Teknik Elektro dengan judul :

“Perancangan Human Machine Interface Siemens Mini CNC MTU 200 Untuk Keperluan Pengawasan Dan Maintenance”

Dengan terselesainya Tugas Akhir ini Penulis mengucapkan terima kasih yang sebesar-besarnya kepada beberapa pihak. Yang pertama untuk kedua orang tua yang senantiasa mendoakan dan memberikan dukungan dengan tulus tiada henti. Bapak Ir. Joko Susila, M.T. selaku Kepala Departemen Teknik Elektro Otomasi. Bapak Imam Arifin S.T., M.T. selaku Kepala Laboratorium Automation and Computer System Laboratory. Bapak Slamet Budiprayitno, S.T., M.T. dan Bapak Moh.Abdul Hady, S.T., M.T selaku dosen pembimbing yang telah memberikan arahan dan bimbingannya. Serta semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Tugas Akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Tugas Akhir ini. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, Juli 2018

Penulis

-----Halaman ini sengaja dikosongkan-----

DAFTAR ISI

	HALAMAN
HALAMAN JUDUL.....	i
PERNYATAAN KEASLIAN TUGAS AKHIR.....	v
ABSTRAK.....	ix
<i>ABSTRACT</i>	xi
KATA PENGANTAR	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL.....	xix
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Permasalahan	2
1.3 Batasan Masalah	3
1.4 Tujuan	3
1.5 Metodologi Penelitian	3
1.6 Sistematika Laporan.....	4
BAB II TEORI DASAR.....	7
2.1 Mesin CNC	7
2.2 XML.....	8
2.3 PLC	8
2.4 Sinumerik 808D	16
2.5 Maintenance	17
BAB III PERANCANGAN <i>HARDWARE</i> DAN <i>SOFTWARE</i>.....	19
3.1. Perancangan Sistem	19
3.2. Diagram <i>Wiring</i> CNC	20
3.3. Pemograman XML.....	21
3.4. Simulasi Program	22
BAB IV PENGUJIAN DAN ANALISA	25
4.1 Pengujian Simulasi Program	25
4.2 Pengujian Kecepatan Motor Pada Mini CNC	26
BAB V PENUTUP.....	31
5.1 Kesimpulan	31

5.2 Saran	31
DAFTAR PUSTAKA.....	33
LAMPIRAN	1
LAMPIRAN B.....	1

DAFTAR GAMBAR

	HALAMAN
Gambar 2.1 Mesin CNC <i>Milling</i>	7
Gambar 2.2 PLC	15
Gambar 2.3 Sinumetik 808D.....	16
Gambar 2.4 Diagram <i>Wiring</i>	17
Gambar 3.1 Diagram Fungsional Cara Kerja HMI	19
Gambar 3.2 Diagram Alur.....	20
Gambar 3.3 <i>Wiring</i> Sinumerik	20
Gambar 3.4 Pemograman XML untuk Motor Spindle	21
Gambar 3.5 Simulasi Program	22
Gambar 3.6 Flow Chart Program	23
Gambar 4.1 Hasil Simulasi	25
Gambar 4.2 Program Alarm.....	26
Gambar 4.3 Kecepatan Spindle saat 20 RPM	27
Gambar 4.4 Kecepatan Spindle 4000 Rpm	28
Gambar 4.5 Kecepatan Spindle saat 400 RPM	29
Gambar 4.6 Program Parameter	30

-----Halaman ini sengaja dikosongkan-----

DAFTAR TABEL

	HALAMAN
Tabel 3.1 List Address Database.....	22
Tabel 4.1 Hasil Pengujian Kecepatan Spindle	27

-----Halaman ini sengaja dikosongkan-----

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan Industri modern belakangan ini sangatlah berkembang dengan banyaknya mesin baru dan pengembangan teknologi yang selalu update menyebabkan banyak perubahan dalam perkembangan industri. Teknologi yang berkembang ini menciptakan banyak peluang bagi pengusaha untuk menciptakan bisnis kreatif dengan menggunakan banyak teknologi yang ada. Sehingga Perkembangan pengetahuan teknologi tentang mesin industri sangat dibutuhkan.

PT. CNC CONTROLLER INDONESIA merupakan perusahaan yang bergerak dalam bidang manufaktur dan restorasi mesin CNC (Computer Numerical Control). Perusahaan ini memiliki produk berupa mesin Mini CNC MTU 200 yang digunakan sebagai pembelajaran bagi sekolah atau institusi untuk mempelajari mesin industri. Mesin ini memiliki kemampuan yang disamakan dengan Mesin CNC yang asli, dengan spesifikasi yang lebih rendah untuk memudahkan sebagai latihan atau pembelajaran.

Penggunaan Mesin CNC sekarang sangat dibutuhkan bagi industri. Mesin ini dapat digunakan untuk membuat benda yang sangatlah detail dan rinci. Dengan ketelitian mencapai 0,1 cm membuat mesin ini sangat fleksibel di dunia industri. Mesin CNC juga disebut mesin induk karena digunakan untuk membuat komponen dari mesin lain seperti baut dan mur. Untuk mesin CNC biasanya menggunakan *spindle* yang bermacam-macam untuk ketelitiannya.[1] Untuk itu diperlukan banyak latihan agar dapat menggunakan mesin ini

Mesin Mini CNC ini diharapkan dapat digunakan sebagai anak muda untuk sarana latihan sebelum masuk di dunia industri. Pembelajaran dari pembuatan mesin sampai cara pemakaian mesin diperlukan untuk dapat cara untuk menggunakan mesin CNC sangat juga diperlukan. Pemakaian CNC sekarang sudah meluas di dunia dan sudah banyak macamnya seperti laser cutting, 3D printing, Mesin bubut, mesin *milling* dan lain sebagainya.[2]

Mesin Mini CNC buatan perusahaan PT.CNC CONTROLLER INDONESIA memiliki spesifikasi yang sangat cukup untuk fokus pembelajaran. Tidak hanya memiliki spesifikasi yang sama Mini CNC ini dapat dioperasikan menggunakan Komputer yang juga disediakan

oleh Perusahaan. Perusahaan ini adalah menyediakan sistem kerja praktek untuk mahasiswa dan siswa SMK untuk bisa menambah wawasan pada dan membantu pada pengerjaan permesinan. Perusahaan ini dibagi 2 divisi: elektrik dan mekanik. Divisi memiliki fokus masing masing untuk pengerjaannya pada divisi mekanik bekerja untuk membuat dan mensetting motor. Dan divisi elektrik berfungsi untuk panel. Bagian elektrik digunakan untuk memasang dan *wiring* bagian panel sampai *controller*. Juga memiliki kewajiban untuk memberi layanan service pada client jika ada masalah pada wiring dan program. Dengan ada masalah service ini biasanya memakan waktu yang lama karena diperlukannya pengecekan pada mesin secara menyeluruh.

Waktu terbuang untuk pengecekan secara menyeluruh menyebabkan adanya alasan untuk membuat judul tugas akhir ini agar dapat menyingkat waktu pada saat bagian elektrik memberi layanan service pada *client*. Pada tugas akhir ini akan dibuat Human Machine Interface (HMI) untuk mesin Mini CNC MTU 200 sebagai Monitoring Mesin Menggunakan Sinumerik 808D. Menggunakan bahasa pemrograman Xml yang digunakan khusus untuk Sinumerik 808D untuk menampilkan data yang ada dalam mesin Mini CNC 808D

Dari tugas akhir ini diharapkan dapat membantu perusahaan PT.CNC CONTROLLER INDONESIA dengan data yang ada dalam mesin dengan pembacaan melalui data servo drive yang dikirimkan kepada masing-masing komponen. Pada saat ada kerusakan pada mesin ini dapat terbaca secara cepat dan efektif. Sehingga saat mesin pada klien mengalami kerusakan perusahaan dapat membaca masalah apa yang ada dalam mesin Mini CNC MTU 200 tanpa menghabiskan banyak waktu.

1.2 Permasalahan

Agar pembahasan dari laporan yang ditulis ini menerangkan pokok masalah secara jelas dan terperinci, maka diberikan batasan-batasan masalah yang melingkupi. Beberapa komponen sering rusak karena penggunaan yang terlalu sering seperti contoh motor Servo, PLC, dan Mata Pahat. Mesin CNC tidak memiliki kemampuan untuk pembacaan pada komponennya. Banyaknya waktu yang terbuang untuk melakukan identifikasi pada komponen yang rusak karena tidak adanya sistem yang mengawasi.

1.3 Batasan Masalah

Dalam pembuatan alat Tugas Akhir ini ada batasan masalah yaitu Mesin CNC yang siap digunakan percobaan hanya memiliki 1 motor Servo untuk bisa digunakan datanya

1.4 Tujuan

Tujuan yang akan dicapai dari tugas akhir ini adalah membuat HMI untuk mengawasi Komponen yang ada pada mesin Mini CNC MTU 200 seperti Motor Servo.

1.5 Metodologi Penelitian

Dalam pelaksanaan Tugas Akhir yang Pembuatan HMI Siemens Mini CNC MTU 200 , ada beberapa kegiatan yang dapat diuraikan sebagai berikut :

A. Tahap persiapan

Pada tahap ini dilakukan study literature mengenai cara kerja Mesin CNC, Program XML, PLC, cara kerja HMI Sinumerik 808D, Servo Drive

B. Tahap Identifikasi dan Permodelan Sistem

Pada tahap ini dilakukan identifikasi dari system alat sesuai dengan data yang telah didapatkan dari studi literatur serta dilakukan permodelan dari alat yang akan dikerjakan.

C. Tahap Perancangan

Pada tahap ini akan dilakukan sebuah perancangan dan permodelan dari alat tersebut, Pada tahap ini akan pada Mesin Mini CNC akan ditambahkan pemrograman XML. Kecepatan Motor Servo dapat dibaca dan ditampilkan ke dalam HMI Sinumerik 808D. Kemudian pada bagian halaman pengawasan akan diberikan list bagian apa saja yang diawasi di mesin Mini CNC

Setelah mendapat teori dasar yang mendasari pembuatan Tugas Akhir ini, baru dilakukan perancangan alat Tugas Akhir ini. Perencanaan dilakukan agar alat yang dibuat sesuai dengan teori dasar yang dimiliki dengan menerapkan ke dalam praktik bertujuan untuk alat yang dibuat memiliki hasil yang sesuai dengan yang diharapkan. Perancangan yang dilakukan terdiri dari perancangan *hardware* yang meliputi perancangan rangkaian elektronika, perancangan mekanik dan perancangan *software*.

D. Tahap Pembuatan Alat

Pada tahap ini akan dilakukan pembuatan alat sesuai perancangan yang dibuat, berdasarkan data yang telah dikumpulkan melalui studi literatur

E. Tahap Pengujian Alat

Pada tahap ini akan dilakukan pengujian alat, menganalisa kesalahan atau kegagalan pada alat dan mengatasi permasalahan tersebut. Pada tahap ini, menganalisa faktor apa saja yang menyebabkan alat tidak bekerja sesuai dengan keinginan atau terjadi *error*. Tahapan ini dilakukan berdasarkan urutan di bawah ini :

- Pengujian Perputaran Motor servo
- Pengujian Pembacaan dan Penampilan data motor di sinumerik 808D

F. Tahap Penyusunan Laporan

Setelah alat berhasil dibuat dan berkerja dengan baik tanpa adanya *error*, pengambilan data dan analisa data terpenuhi, maka tahap selanjutnya yaitu penyusunan laporan untuk buku Tugas Akhir. Diharapkan buku Tugas Akhir ini bermanfaat bagi semua orang, dan dapat dijadikan pedoman dalam melanjutkan dan mengembangkan ide Tugas Akhir ini

1.6 Sistematika Laporan

Pembahasan Tugas Akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut:

Bab I Pendahuluan

Membahas tentang latar belakang, perumusan masalah, batasan masalah, maksud dan tujuan, sistematika laporan, metodologi, serta relevansi Tugas Akhir yang dibuat.

Bab II Teori Dasar

Menjelaskan teori yang berisi teori-teori dasar yang dijadikan landasan dan mendukung dalam perencanaan dan pembuatan alat yang dibuat.

Bab III Perancangan Alat

Membahas perencanaan dan pembuatan tentang perencanaan dan pembuatan hardware yang meliputi desain mekanik dan perancangan software yang meliputi program yang akan digunakan untuk menjalankan alat tersebut.

Bab IV Pengukuran dan Analisa

Membahas pengujian alat dan menganalisa data yang didapat dari pengujian tersebut serta membahas tentang pengukuran, pengujian, dan penganalisaan terhadap alat.

Bab V Penutup

Berisi penutup yang menjelaskan tentang kesimpulan yang didapat dari Tugas Akhir ini dan saran-saran untuk pengembangan alat ini lebih lanjut.

-----Halaman ini sengaja dikosongkan-----

BAB II TEORI DASAR

Beberapa teori penunjang yang dipaparkan dalam buku Tugas Akhir ini adalah teori dasar mengenai *Maintenance*, Mesin CNC, Pemrograman XML, PLC, HMI Sinumerik 808D

2.1 Mesin CNC

alat industri yang digunakan untuk mesin bubut dan mesin *milling*. CNC ini biasa digunakan untuk membuat dan membentuk benda untuk hal yang sangat detail seperti mur, baut dan bidak catur. Bahkan untuk mesin CNC skala besar bisa digunakan pembuatan bodi dan bagian dalam kapal mesin CNC menggunakan bahasa pemrograman *G-code* yaitu bahasa pemrograman baku yang digunakan oleh mesin induk dan biasanya *G code* digunakan untuk mengatur motor dari arah, kecepatan, dan jalur yang digunakan.[1] Mesin Mini CNC MTU 200 yang ada pada Gambar 2.1 juga menggunakan bahasa *G-code* karena memang fungsi mesin ini digunakan sebagai pembelajaran dari mesin yang ada di industri.



Gambar 2.1 Mesin CNC *Milling*

Mesin CNC pertama kali digunakan dalam proses mengukir, membuat lubang, memutar, mengasah, memotong, membengkokkan pipa dan membuat berbagai bentuk.[3] Dibandingkan dengan peralatan

biasa, mesin yang dikontrol dengan kode angka ini lebih cermat, cepat, konsisten dan fleksibel, bahkan untuk *manufacturing* yang rumit. Rancangan produk dapat diubah atau disesuaikan cukup dengan mengubah instruksi saja.[3]

Pada sistem mesin mini CNC, titik awal penempatan alat potong yang digunakan sebagai acuan adalah menetapkan titik referensi yang berlaku tetap selama proses operasi mesin berlangsung. Untuk mesin bubut, titik referensinya diletakkan pada sumbu (pusat) benda kerja yang akan dikerjakan pada bagian ujung. Sedangkan pada mesin *frais*, titik referensinya diletakkan pada pertemuan antara dua sisi pada benda kerja yang akan dikerjakan.

2.2 XML

Penggunaan HMI Sinumerik bisa diprogram menggunakan pemrograman XML. terletak pada inti *web service*, yang digunakan untuk mendeskripsikan data.[4] Fungsi utama dari XML adalah komunikasi antar aplikasi, integrasi data, dan komunikasi aplikasi eksternal dengan *partner* luaran. Dengan standarisasi XML, aplikasi-aplikasi yang berbeda dapat dengan mudah berkomunikasi antar satu XML adalah singkatan dari *eXtensible Markup Language*. [4] Bahasa *markup* adalah sekumpulan aturan-aturan yang mendefinisikan suatu sintaks yang digunakan untuk menjelaskan, dan mendeskripsikan teks atau data dalam sebuah dokumen melalui penggunaan *tag*. Bahasa *markup* lain yang populer seperti HTML, menggambarkan kepada *browser web* tentang bagaimana menampilkan format teks, data, dan grafik ke layar komputer ketika sedang mengunjungi sebuah situs *web*.

XML merupakan sebuah bahasa *markup* yang digunakan untuk mengolah meta data (informasi tentang data) yang menggambarkan struktur dan tujuan data yang terdapat dalam dokumen XML, tetapi bukan menggambarkan format tampilan data tersebut. XML adalah sebuah standar sederhana yang digunakan untuk mendeskripsikan data teks dengan cara *self-describing* (deskripsi diri). XML juga dapat digunakan untuk mendefinisikan domain tertentu lainnya, seperti musik, matematika, keuangan dan lain-lain yang menggunakan bahasa *markup* terstruktur.

2.3 PLC

Penggunaan alat yang digunakan pada industri sekarang kebanyakan PLC adalah rangkaian alat elektronik atau kontroler yang

dapat mengerjakan berbagai fungsi – fungsi *control* pada level yang kompleks. PLC awal dirancang untuk menggantikan sistem logika relai. PLC ini diprogram dalam "logika *ladder*", yang sangat menyerupai diagram skematik logika *relay*. Notasi program ini dipilih untuk mengurangi tuntutan pelatihan bagi teknisi yang ada. PLC awal lainnya menggunakan bentuk pemrograman daftar instruksi, berdasarkan pada pemecah logika berbasis tumpukan. PLC dapat di program dan di control oleh operator atau admin melalui komputer. Program yang ada didalam PLC dinamakan program diagram ladder. Program ini menggambarkan hubungan yang diperlukan untuk suatu proses. PLC akan mengoperasikan system I/O yang dapat terdiri dari kontak dan coil yang tersambung dengan sensor dan actuator. [5]

Pada awalnya PLC bekerja untuk kondisi ON dan OFF untuk menggerakan motor, sensor, dan actuator. PLC juga merupakan suatu system elektronika digital yang dirancang agar dapat mengendalikan mesin dengan proses mengimplementasikan fungsi kendali sekuensial, operasi, perwaktuan (*timing*), pencacahan (*counting*), dan aritmatika. Kontroler ini juga memiliki I/O, unit *memory*, unit *control*, dan *processor* sendiri sehingga dapat disebut dengan komputer digital. [2]

Mereka dapat dirancang untuk pengaturan ganda I/O digital dan analog, rentang suhu yang diperpanjang, kekebalan terhadap gangguan listrik, dan ketahanan terhadap getaran dan benturan. Program untuk mengontrol operasi mesin biasanya disimpan dalam memori yang didukung baterai atau *non-volatile*. PLC, kontrol, *sequencing*, dan logika pengaman *interlock* untuk manufaktur mobil terutama terdiri dari *relay*, *timer cam*, *sequencer drum*, dan pengontrol *loop* tertutup khusus. Karena ini bisa berjumlah ratusan atau bahkan ribuan, proses untuk memperbaiki fasilitas-fasilitas tersebut untuk perubahan model tahunan sangat memakan waktu dan mahal, seperti yang diperlukan para tukang listrik untuk secara individu *re-wire relay* untuk mengubah karakteristik operasional mereka. Ketika komputer digital sudah ada, menjadi perangkat yang dapat diprogram untuk tujuan umum, mereka segera diterapkan untuk mengontrol logika sekuensial dan kombinatorial dalam proses industri. Namun, komputer awal ini membutuhkan pemrogram khusus dan kontrol lingkungan operasi yang ketat untuk suhu, kebersihan, dan kualitas daya.[3]

Untuk memenuhi tantangan ini, PLC dikembangkan dengan beberapa atribut kunci. Ini akan menoleransi lingkungan kerja, itu akan mendukung masukan dan keluaran yang terpisah (*bit-form*) dengan

cara yang mudah diperluas, tidak akan membutuhkan bertahun-tahun pelatihan untuk digunakan, dan itu akan memungkinkan operasinya untuk dimonitor. Karena banyak proses industri memiliki skala waktu yang mudah diatasi oleh waktu respons milidetik, elektronik modern (cepat, kecil, andal) sangat memfasilitasi pembuatan pengendali yang andal, dan kinerja dapat diperdagangkan untuk keandalan. PLC modern dapat diprogram dalam berbagai cara, mulai dari logika tangga yang diturunkan ke Bahasa pemrograman seperti *basic* dan C.

Metode lainnya adalah *logic state*, bahasa pemrograman tingkat tinggi yang dirancang untuk memprogram PLC berdasarkan diagram transisi negara. Mayoritas sistem PLC saat ini mematuhi standar pemrograman sistem kontrol IEC 61131/3 yang mendefinisikan 5 bahasa: *Ladder Diagram* (LD), *Structured Text* (ST), *Function Blok Diagram* (FBD), *Instruction List* (IL) dan *Sequential Flow Chart* (SFC). Banyak PLC lama yang tidak memiliki terminal pemrograman yang menyertainya yang mampu representasi grafis dari logika, dan jadi logika itu diwakili sebagai serangkaian ekspresi logika dalam beberapa versi format *Boolean*, mirip dengan aljabar *Boolean*. [7]

Ketika terminal pemrograman berevolusi, itu menjadi lebih umum untuk logika *ladder* untuk digunakan, untuk alasan yang disebutkan di atas dan karena itu adalah format yang umum digunakan untuk panel kontrol elektromekanik. Format yang lebih baru seperti logika negara dan Blok Fungsi (yang mirip dengan cara logika digambarkan ketika menggunakan sirkuit logika terintegrasi digital) ada, tetapi mereka masih tidak sepopuler logika tangga. Alasan utama untuk ini adalah bahwa PLC memecahkan logika dalam urutan diprediksi dan berulang, dan logika tangga memungkinkan *programmer* (orang yang menulis logika) untuk melihat masalah dengan waktu urutan logika lebih mudah daripada yang mungkin di lain format. Untuk menjalankan suatu PLC pasti memerlukan program untuk membuat PLC tersebut bekerja, program ini disebut *Programming Devices* yaitu alat untuk memprogram suatu peralatan yang digunakan untuk menuliskan, mengedit, memodifikasi atau *monitoring* program yang ada di dalam memori PLC.[6]

Sistem pemrograman PLC dapat dilakukan dengan dua cara. Pertama dengan menggunakan *personal computer*, yaitu rancangan kontrol konvensional ditulis dalam bentuk *ladder diagram* dan dapat langsung didownload ke PLC. Kedua dengan menggunakan

programming console, yaitu *ladder* diagram yang telah didapat tadi diubah terlebih dahulu menjadi mnemonic code dan selanjutnya dapat dituliskan pada *keypad* dari *console*. PLC mengenal berbagai macam perangkat lunak, termasuk *State Language*, *SFC*, dan bahkan *C*. Yang paling populer digunakan ialah *RLL (Relay Ladder Logic)*[2].

Semua bahasa pemrograman tersebut dibuat berdasarkan proses sekuensial yang terjadi dalam *plant* (sistem yang dikendalikan). Semua instruksi dalam program akan dieksekusi oleh modul CPU, dan penulisan program itu bisa dilakukan pada keadaan *online* maupun *offline*. Jadi PLC dapat bisa ditulis program kontrol pada saat ia mengendalikan proses tanpa mengganggu pengendalian yang sedang dilakukan. Eksekusi perangkat lunak tidak akan mempengaruhi operasi I/O yang tengah berlangsung. PLC sendiri memiliki struktur datanya sendiri yaitu:

a. *CPU (Central Processing Unit)*

Untuk mengontrol dan mengawasi semua pengoperasian dalam PLC, melaksanakan program yang sudah tersimpan didalam *memory* PLC itu sendiri. CPU merupakan otak dari *system* PLC. CPU akan bertugas mengambil intruksi memori, mengkodennya, dan kemudian mengeksekusi intruksi tersebut. Selama proses tersebut, *processor* akan menghasilkan sinyal kontrol yang memindahkan data I/O *port* atau sebaliknya, melakukan fungsi aritmatika dan logika juga mendeteksi sinyal dari luar CPU. CPU mempunyai elemen kontrol yang disebut *Arithmetic and Logic Unit (ALU)*, sehingga mampu mengerjakan operasi logika (penjumlahan, pengurangan, perkalian, pembagian, dan lain – lain) dan aritmatika (operasi OR, AND, NOT, dan lain – lain). [4]

Sebagian besar CPU berisi baterai cadangan yang berfungsi untuk menjaga program operasi yang ada dalam penyimpanan jika terjadi kegagalan *supply* daya pada PLC. Pada PLC tertentu kadang kita jumpai pula beberapa prosesor sekaligus dalam satu modul, yang ditujukan untuk mendukung keandalan sistem. Beberapa prosesor tersebut bekerja sama dengan suatu prosedur tertentu untuk meningkatkan kinerja pengendalian. [7]

b. *Memory*

Tempat penyimpanan data dalam PLC. Memori yang terdapat pada PLC berfungsi untuk menyimpan program dan memberikan

lokasi-lokasi dimana hasil-hasil perhitungan dapat disimpan di dalamnya. Memori ini umumnya menjadi satu modul dengan *processor*/CPU. Jika berbentuk memori eksternal maka itu merupakan memori tambahan. [7]

Data yang tersimpan di dalam memori berupa *Operating System* PLC, Status *Input – Output*, Data memori, dan Program yang dibuat. Dalam *memory* PLC ada yang disebut dengan *Operating System Memory*, Data (status) memori, dan program memori. *Operating System Memory* yang berfungsi untuk menyimpan *Operating System* PLC. Memori ini berupa ROM (*Read Only Memory*) sehingga tidak dapat diubah oleh *user*. Data (status) memori berfungsi untuk menyimpan status *input-output* tiap saat. Memori ini berupa RAM (*Random Access Memory*) sehingga dapat berubah sesuai kondisi *input-output*. Status akan kembali ke kondisi awal jika PLC mati. Program memori berfungsi untuk menyimpan program pengguna. Jenis memori ini berupa RAM.

Komponen ini menggunakan baterai *backup* untuk menyimpan program selama jangka waktu tertentu. Selain itu dapat berupa EEPROM (*Electrical Erasable Programmable Read Only Memory*), yaitu jenis ROM yang dapat diprogram dan dihapus oleh user. Sedangkan untuk kebutuhan program oleh pengguna, area memori PLC ada *register*, *flag register*, *Auxiliary relays*, *timer*, dan *counter*. *Register* berfungsi untuk menyimpan sekumpulan bit data, baik berupa *nibble* (4 bit), *byte* (8 byte), maupun *word* (16 bit). *Flag register* berfungsi untuk mengindikasikan perubahan kondisi (*state*) *input-output* fisik.

Flag register berupa satu bit data. CPU pada umumnya memiliki *internal flag* untuk berbagai keperluan internal PLC. *Auxiliary relays* adalah elemen memori 1bit dalam RAM yang digunakan untuk memanipulasi data dalam program. *Auxiliary relays* disebut juga relay yang imajiner, karena dapat menggantikan fungsi *relay* namun berbentuk program. Timer adalah pemberian penundaan waktu dalam suatu proses. Pada umumnya Timer memiliki alamat khusus. Counter adalah komponen penghitung *input* pulsa yang diberikan *input device*. CPU memiliki counter internal. Counter ini umumnya memiliki alamat khusus. Memori juga memiliki jenis jenis seperti RAM yang merupakan memori yang cepat dan bersifat volatile (data yang hilang bila arus listrik mati). RAM digunakan sebagai memori utama dalam PLC, berfungsi sebagai tempat penyimpanan

program, program yang telah ditulis tersimpan di dalam RAM yang ada di dalam PLC sehingga dapat diubah/diedit melalui programming unit, namun kerugian penyimpanan di RAM adalah program dan data akan hilang ketika power supply mati, dan untuk mengatasi hal ini RAM dapat di backup dengan baterai, sehingga meskipun power supply mati, program dan data tidak hilang, umumnya bila baterai tidak rusak, program dan data bisa di simpan selama bertahun-tahun. Lalu ada ROM (*Read Only Memory*), merupakan tempat penyimpanan Operating System yang dibuat oleh pabrik pembuat PLC. Operating System ini hanya dapat dibaca oleh processor dan berfungsi untuk mengeksekusi program yang tersimpan di dalam RAM.[7]

Ketiga ada EPROM (*Erasable Programmable Read Only Memory*) adalah jenis memori yang cepat dan juga murah harganya, sama dengan memori RAM hanya saja EPROM bersifat *non-volatile*, artinya isi memori ini tetap ada walaupun supply tegangan hilang. Untuk keperluan modifikasi program maka memori ini harus dikosongkan isinya melalui penyinaran dengan sinar ultraviolet. Karena begitu kompleksnya proses penghapusan untuk memprogram ulang bahkan meskipun harganya murah, orang cenderung memilih RAM. Baru bila programnya sudah benar dan lengkap langsung bisa ditransfer ke EPROM secara permanen. Keempat ada EEPROM adalah memori yang mirip dengan memori EPROM, hanya saja untuk proses penghapusannya menggunakan arus listrik. [7]

Kapasitas memori tergantung penggunaannya dan seberapa jauh Anda sebagai mengoptimalkan ruang memori PLC yang Anda miliki, yang berarti pula tergantung seberapa banyak lokasi yang diperlukan program kontrol untuk mengendalikan plant tertentu. Program kontrol untuk pengaliran bahan bakar dalam turbin gas tentu membutuhkan lokasi memori yang lebih banyak dibandingkan dengan program kontrol untuk menggerakkan putaran mekanik robot pemasang bodi mobil pada industri otomotif. Suatu modul memori tambahan bisa juga diberikan ke sistem utama apabila kebutuhan memori memang meningkat[7]

c. Input/Output Module

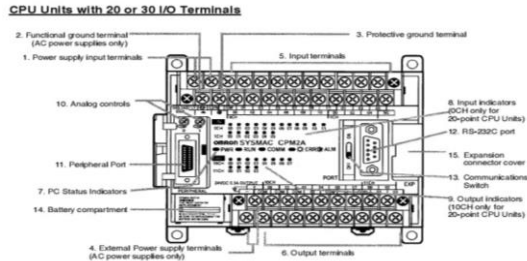
Suatu peralatan atau perangkat elektronik yang berfungsi sebagai perantara atau penghubung (*interface*) antara CPU dengan peralatan *input output* devices. Modul I/O bertugas mengatur hubungan PLC dengan piranti eksternal atau periferal yang bisa berupa suatu komputer host, saklar-saklar, unit penggerak motor, dan berbagai

macam sumber sinyal yang terdapat dalam plant. Fungsi dari sebuah modul *input* adalah untuk mengubah sinyal masukan dari sensor ke PLC untuk diproses dibagian CPU. Sedangkan modul *output* adalah kebalikannya, mengubah sinyal PLC kedalam sinyal yang sesuai untuk menggerakkan aktuator. Dari modul *input* dan *output* kita dapat menentukan jenis suatu PLC dari hubungan antara processor dengan modul *input* dan *output* yaitu Compact PLC adalah bila *input* modul processor dan *output* modul dikemas dalam suatu wadah dan Modular PLC yang bila modul *input*, modul *output* dan processor dikemas secara tersendiri. Modul *input* adalah modul tempat menghubungkan sensor-sensor dengan modul itu sendiri. Sinyal sensor tersebut selanjutnya akan diteruskan ke Processor.[7]

Modul *input* berfungsi untuk mengkonversi sinyal diskrit dan analog yang berasal dari *input* device menjadi sinyal digital. Modul *input* bertugas untuk menerima sinyal dari unit pengindera perifer, dan memberikan pengaturan sinyal, terminasi, isolasi, maupun indikator keadaan sinyal masukan. Sinyal-sinyal dari piranti perifer akan di-scan dan keadaannya akan dikomunikasikan melalui modul antar muka dalam PLC. Modul *input* dibagi menjadi dua yaitu *input* analog yang dapat menerima tegangan dan arus dengan level tertentu (misal 0–10V, 4–20mA) dari *input* device analog (misal: sensor analog, potensiometer) dan digital yang berfungsi untuk menghubungkan *input* diskrit fisik (switch, sensor) dengan PLC. Modul ini tersedia dalam tegangan DC dan AC (umumnya: 240VAC, 120VAC, 24VDC, dan 5VDC). Didalamnya terdapat optoisolator untuk mencegah lonjakan tegangan tinggi masuk PLC (sebagai pengaman). *Output* Modules berfungsi untuk mengkonversi sinyal digital menjadi sinyal analog sehingga dapat mengaktifkan kembali peralatan luar (*output* devices). Modul *output* mengeluarkan sinyal dari Processor ke kontrol elemen yang diperlukan untuk menggerakkan aktuator sesuai dengan tugas yang telah diberikan.

Modul keluaran mengaktifasi berbagai macam piranti seperti aktuator hidrolik, pneumatik, solenoid, starter motor, dan tampilan status titik-titik perifer yang terhubung dalam sistem. Fungsi modul keluaran lainnya mencakup conditioning, terminasi dan juga pengisolasian sinyal-sinyal yang ada. Proses aktivasi itu tentu saja dilakukan dengan pengiriman sinyal-sinyal diskret dan analog yang relevan, berdasarkan watak PLC sendiri yang merupakan piranti digital. Modul *output* juga dibagi menjadi dua yaitu *output* analog yang

dapat memberikan tegangan dan arus dengan level tertentu (misal 0–10V, 4–20 mA) pada *output* device analog (misal: motor DC, motor AC, control valve) dan modul *output* digital yang menghubungkan *output* diskrit fisik (lampu, relay, solenoid, motor) dengan PLC.



Gambar 2.2 PLC

PLC CJ1M merupakan PLC keluaran dari Omron dengan seri CJ. PLC ini merupakan PLC tipe modular yang berarti I/O nya bisa ditambah sesuai dengan kebutuhan pengguna dan komponennya terpisah pisah. CJ1M ini terpisah pisah menjadi 4 bagian yaitu CJ1M CPU 13 yang merupakan bagian yang berfungsi sebagai CPU atau *Prosesor* dari PLC Seri CJ ini. CPU 13 ini memiliki kapasitas memory sebesar 20Ksteps dan mempunyai I/O hingga 640 point. [6]

Untuk komunikasi, PLC ini menggunakan serial RS-232 dan Ethernet karena tersedia port Peripheral Port untuk Ethernet dan RS-232 Port untuk serialnya. Untuk programnya menggunakan Software bernama CX Programmer. Bagian kedua merupakan CJ1M PA-202 yang merupakan bagian Power Supply untuk unit CPU. CJ1M PA-202 memiliki spesifikasi tegangan sebesar 110 – 240 VAC dan total daya 14 watt. PA-202 memiliki *Output* untuk 5 VDC sebesar 2,8 A dan untuk 24 VDC sebesar 0,4 VDC. PA-202 memiliki range sebesar 50 Hz – 60 Hz. Bagian ketiga ada CJ1M ID-211 bagian yang merupakan merupakan modul *input* dari PLC. ID-211 ini merupakan *input* digital yang tidak memiliki spesifikasi untuk sinyal digital namun tidak bisa digunakan untuk sinyal analog. PLC ini memiliki *input* sebanyak 16, voltase sebesar 24 VDC dan arus sebesar 7mA. Bagian keempat merupakan CJ1M OC-211 yang merupakan modul *output* dari PLC. OC-211 ini hanya menerima hasil keluaran seperti ON/OFF

atau I/O karena OC-211 hanya digunakan untuk mengontrol device external saja. Modul ini terdiri dari 16 *output* saja serta memiliki voltase 12 -24 VDC dan maksimal arus 0,5 A per point dan 5 A per Unit.

2.4 Sinumerik 808D

Perangkat ini merupakan setpoint yang digunakan pada mesin Mini CNC MTU 200. Komponen ini merupakan buatan perusahaan Siemens yang memiliki spesialisasi untuk mesin CNC karena controller CNC harus memiliki kemampuan untuk mengubah G-Code menjadi gerak yang deprogram untuk motor Servo.



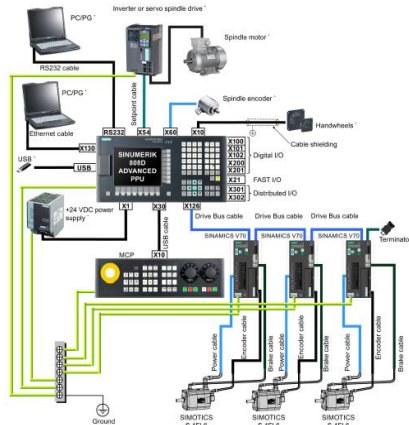
Gambar 2.3 Sinumerik 808D

Program G-Code merupakan program yang digunakan untuk menjalankan mesin dengan berbasis angka. Pemrograman G-Code ini biasa digunakan untuk menyatakan gerakan yang akan dilakukan oleh mesin. Pemrograman G-Code memiliki 2 metode yaitu :

- Metode Inkremental Adalah suatu metode pemrograman dimana titik referensinya selalu berubah, yaitu titik terakhir yang dituju menjadi titik referensi baru untuk ukuran berikutnya.
- Metode Absolut Adalah suatu metode pemrograman dimana titik referensinya selalu tetap yaitu satu titik/tempat dijadikan referensi untuk semua ukuran berikutnya.[1][5]

Penggunaan sinumerik 808D sangat penting untuk mesin CNC merupakan alat yang digunakan untuk memasukan program yang dibutuhkan untuk menjalankan mesin dan digunakan untuk menunjukkan status mesin pada mesin CNC. Dengan ada sinumerik ini

diharapkan bisa lebih mudah untuk menjalankan program. Karena penggunaan sinumerik ini selaras karena beberapa komponen seperti PLC, Motor Servo, Servo Drive juga menggunakan produk milik Siemens.[7]



Gambar 2.4 Diagram Wiring

Perangkat ini menghubungkan banyak komponen pada mesin Mini CNC seperti pada Gambar 2.4 menunjukkan bahwa perangkat ini digunakan sebagai set point pada mesin CNC untuk memasukkan dan membuat mesin menjalankan program dari G-Code yang dimasukkan pada mesin Sinumerik 808D.[6]

2.5 Maintenance

Pengawasan merupakan hal penting dalam sebuah perusahaan yang bekerja dalam bidang apapun. Perusahaan di bidang *manufacturing* biasanya membutuhkan *maintenance* yang digunakan sebagai pengurangan dana untuk mereparasi barang yang digunakan dan dimiliki oleh perusahaan.[8]

Kegiatan Maintenance (perawatan) secara garis besar dilakukan untuk mencegah kerusakan mesin atau peralatan yang digunakan untuk kegiatan produksi terlalu cepat, selain itu kegiatan perawatan haruslah memiliki kriteria efektif, efisien, serta berbiaya rendah.[9]

Pada dasarnya Maintenance memiliki 2 kriteria yang dibagi menurut waktu dari *maintenance*

- *Condition maintenance*, yaitu aktivitas perawatan untuk mempertahankan keadaan mesin/peralatan agar dapat berfungsi dengan baik sesuai dengan usia ekonomis mesin itu.
- *Replacement maintenance*, yaitu aktivitas perawatan untuk perbaikan dan penggantian komponen mesin tepat pada waktunya sesuai dengan jadwal yang telah direncanakan.[10]

Kegiatan *Maintenance* (perawatan) secara garis besar dilakukan untuk mencegah kerusakan mesin/peralatan yang digunakan untuk kegiatan produksi terlalu cepat, selain itu kegiatan perawatan haruslah memiliki kriteria efektif, efisien, serta berbiaya rendah.[11] Berikut ini beberapa tujuan kegiatan perawatan antara lain :

- Memperpanjang usia pakai dari mesin/peralatan.
- Menjaga fungsi dari mesin/peralatan agar tetap baik.
- Menjamin ketersediaan optimum mesin/peralatan.
- Menjamin kesiapan operasional mesin/peralatan.
- Mengurangi waktu *downtime* dari mesin/peralatan (memaksimalkan ketersediaan (*availability*))
- Menjamin keselamatan *user* mesin/peralatan tersebut.
- Menjamin kepuasan pelanggan

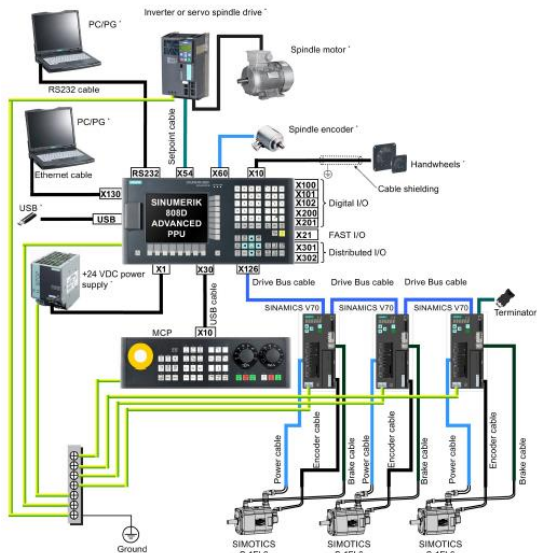
Kegiatan maintenance ini biasa Komponen kritis adalah kondisi suatu komponen yang berpotensi mengalami kerusakan yang berpengaruh pada keandalan operasional unit sistem Penilaian komponen kritis dapat dilakukan dengan menggunakan pendekatan *critical analysis*. Pemeliharaan prediktif sering dirujuk sebagai pemeliharaan berdasarkan kondisi. Artinya, aktifitas pemeliharaan baru dilakukan pada suatu kondisi mesin tertentu.[9] Dalam pemeliharaan prediktif, digunakan berbagai peralatan untuk mendiagnosa mesin untuk mengukur kondisi fisik dari mesin, seperti getaran, suhu, kebisingan, pelumasan, dan korosi. Ketika salah satu parameter ini mencapai kondisi tertentu, aktifitas pemeliharaan dilakukan dengan mengembalikan ke kondisi semula.

BAB III

PERANCANGAN *HARDWARE* DAN *SOFTWARE*

Bab ini membahas perancangan HMI Sinumerik untuk mini CNC MTU 200. Berikut ini bahasan pertama adalah blok diagram dari sistem. Pembahasan kedua adalah wiring HMI dan pemrograman *software*.

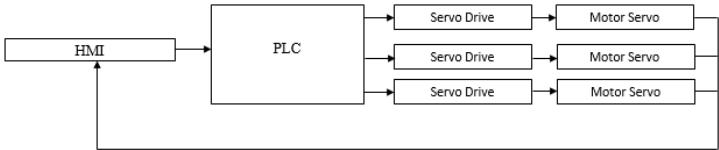
3.1. Perancangan Sistem



Gambar 3.1 Diagram Fungsional Cara Kerja HMI

Dari Gambar 3.1 dapat dilihat Hmi digunakan untuk sebagai set point untuk mesin CNC. Digunakan untuk memasukkan program di HMI lalu menyambungkan ke PLC untuk menjalankan Motor servo dan motor servo memberikan feedback melalui encoder dan langsung ke HMI Sinumerik. Seperti Gambar 3.2 menunjukkan bahwa cara kerja HMI yang ditampilkan secara alur menyambung kepada PLC setelah pemrograman dijalankan dan untuk memanggil *command* pada yang telah di setting di PLC.

PLC akan mengaktifkan command pada servo drive untuk menjalankan motor servo sesuai dengan program yang diinginkan yang telah dimasukan di dalam HMI. Oleh karena itu HMI disebut Setpoint karena program dan setting awal di jalankan SINumerik 808D.

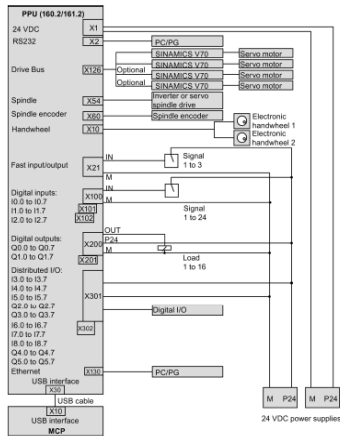


Gambar 3.2 Diagram Alur

Untuk data yang ditampilkan pada maintenance HMI adalah data yang ada dari encoder yang tersambung pada motor servo. Dari Encoder langsung dipasangkan kedalam HMI untuk pembacaan dan menampilkan data. Pada HMI langsung diprogram dengan pemrograman XML untuk menampilkan data tersebut.

3.2. Diagram Wiring CNC

Pada Gambar 3.3 menunjukan Gambar Wiring untuk HMI Sinumerik 808D.



Gambar 3.3 Bagian Dalam Sinumerik

Pada gambar 3.3 ditunjukkan dengan garis yang menyambung pada HMI ini sama dengan wiring yang sudah ditentukan dari *Manual Book*. Pada gambar dijelaskan bahwa Sinumerik 808D bekerja dengan tegangan 24 vdc. Lalu PC disambung ke HMI menggunakan RS232. Untuk penjelasan X50 merupakan alat yang digunakan sebagai protokol paten dari mesin yang berfungsi untuk mendapatkan data dari *encoder*. Untuk HMI ini digunakan untuk menjalankan servo menggunakan *signal pulse*.

3.3. Pemrograman XML

Untuk mengubah *setting* pada HMI bisa menggunakan program XML untuk siemens. Berfungsi sebagai editor pada layar HMI untuk penambahan program dan setting parameter. XML yang digunakan untuk siemens dapat dimasukan pada simulasi.

```
<form name="Menu1">
  <paint>
    <text xpos="213" color="#000000" ypos="39" font="2">CONDITION AXIS X</text>
    <text xpos="38" color="#000000" ypos="199" font="2">TOTAL DISTANCE</text>
    <text xpos="36" color="#000000" ypos="134" font="2">DISTANCE</text>
    <text xpos="39" color="#000000" ypos="75" font="2">SPEED</text>
  </paint>
  <init>
    <caption>user Mask</caption>
    <control name="OutputField1" xpos="42" ypos="236" height="18" width="80" refvar="Nck/Channel/Parameter/30240[0]" hotlink="true" fieldtype="readonly">
    <control name="OutputField2" xpos="42" ypos="168" height="18" width="80" refvar="Nck/Channel/Parameter/33260[0]" hotlink="true" fieldtype="readonly">
    <control name="OutputField3" xpos="46" ypos="108" height="18" width="80" refvar="Nck/Channel/Parameter/34200[0]" hotlink="true" fieldtype="readonly">
    <control name="OutputField1" xpos="42" ypos="236" height="18" width="80" refvar="Nck/Channel/Parameter/32100[2048]" hotlink="true" fieldtype="readonly">
    <control name="OutputField1" xpos="42" ypos="236" height="18" width="80" refvar="Nck/Channel/Parameter/30202[0]" hotlink="true" fieldtype="readonly">
    <control name="OutputField1" xpos="42" ypos="236" height="18" width="80" refvar="Nck/Channel/Parameter/30240[2]" hotlink="true" fieldtype="readonly">
    <control name="OutputField1" xpos="42" ypos="236" height="18" width="80" refvar="Nck/Channel/Parameter/30130[0]" hotlink="true" fieldtype="readonly">
    <control name="OutputField1" xpos="42" ypos="236" height="18" width="80" refvar="Nck/Channel/Parameter/30120[1]" hotlink="true" fieldtype="readonly">
    <control name="OutputField1" xpos="42" ypos="236" height="18" width="80" refvar="Nck/Channel/Parameter/35100[0]" hotlink="true" fieldtype="readonly">
    <control name="OutputField1" xpos="42" ypos="236" height="18" width="80" refvar="Nck/Channel/Parameter/30160[3]" hotlink="true" fieldtype="readonly">
    <control name="OutputField1" xpos="42" ypos="236" height="18" width="80" refvar="Nck/Channel/Parameter/35140[4]" hotlink="true" fieldtype="readonly">
    <control name="OutputField1" xpos="42" ypos="236" height="18" width="80" refvar="Nck/Channel/Parameter/34201[3000]" hotlink="true" fieldtype="readonly">
    <control name="OutputField1" xpos="42" ypos="236" height="18" width="80" refvar="Nck/Channel/Parameter/32500[0]" hotlink="true" fieldtype="readonly">
    <control name="OutputField1" xpos="42" ypos="236" height="18" width="80" refvar="Nck/Channel/Parameter/33440[0]" hotlink="true" fieldtype="readonly">
    <control name="OutputField1" xpos="42" ypos="236" height="18" width="80" refvar="Nck/Channel/Parameter/30240[0]" hotlink="true" fieldtype="readonly">
    <control name="OutputField1" xpos="42" ypos="236" height="18" width="80" refvar="Nck/Channel/Parameter/30270[0]" hotlink="true" fieldtype="readonly">
    <control name="OutputField1" xpos="42" ypos="236" height="18" width="80" refvar="Nck/Channel/Parameter/30710[0]" hotlink="true" fieldtype="readonly">
    <control name="OutputField1" xpos="42" ypos="236" height="18" width="80" refvar="Nck/Channel/Parameter/6010[1000]" hotlink="true" fieldtype="readonly">
    <control name="OutputField1" xpos="42" ypos="236" height="18" width="80" refvar="Nck/Channel/Parameter/4411[0]" hotlink="true" fieldtype="readonly">
  </init>
  <timer />
  <focus_in />
  <edit_changed />
  <index_changed />
  <close />
  <key_event />
  <message />
</form>
```

Gambar 3.4 Pemrograman XML untuk Motor Spindle

Gambar 3.4 menunjukan adress yang dibutuhkan untuk menampilkan data pada HMI menggunakan pemrograman XML yang akan diintegrasikan pada HMI sinumerik. Untuk adress data bisa dilihat pada tabel 3.1 bahwa dengan adress di kiri tabel untuk menampilkan data dan mengisi dalam kurung untuk default value digunakan untuk bisa menampilkan data tersebut. Sehingga dengan untuk memunculkan banyak data juga dibutuhkan banyak program sehingga semakin padat program yang dikeluarkan. Untuk pembacaan program pada mesin dipisah per motor untuk memudahkan

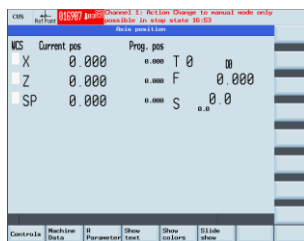
pembacaan. Sehingga jika ada 4 motor di baca. Akan membutuhkan 4 program yang hampir sama untuk per motornya.

Tabel 3.1 List Address Database

No.	Name	Unit	Default value	Description
30130[0]	CTRLOUT_TYPE	-	0	Output type of setpoint
30134[0]	IS_UNIPOLAR_OUTPUT	-	0	Setpoint output is unipolar
30240[0]	ENC_TYPE	-	0	Encoder type of actual value sensing (actual position value)
31020[0]	ENC_RESOL	-	2048	Encoder lines per revolution
31080[1...5]	DRIVE_AX_RATIO_DENUM	-	1	Denominator load gearbox
31080[1...5]	DRIVE_AX_RATIO_NUMERA	-	1	Numerator load gearbox
32020	JOG_VELO	rpm	30	Jog axis velocity
32100	AX_MOTION_DIR	-	1	Traversing direction (not control direction)
32110[0]	ENC_FEEDBACK_POL	-	1	Sign actual value (control direction)
32260[0]	RATED_VELO	rpm	3000	Rated motor speed
34200[0]	ENC_REFP_MODE	-	1	Referencing mode
34210[0]	ENC_REFP_STATE	-	0	Adjustment status of absolute encoder
34220[0]	ENC_ABS_TURNS_MODULO	-	4096	Modulo range for rotary absolute encoder
34230[0]	ENC_SERIAL_NUMBER	-	0	Encoder serial number
35010	GEAR_STEP_CHANGE_ENABLE	-	0	Parameterize gear stage change
35100	SPIND_VELO_LIMIT	rpm	10000	Maximum spindle speed
35110[1...5]	GEAR_STEP_MAX_VELO	rpm	[1]: 500 [2]: 1000 [3]: 2000 [4]: 4000 [5]: 8000	Maximum speed for gear stage change
35120[1...5]	GEAR_STEP_MIN_VELO	rpm	[1]: 50 [2]: 400 [3]: 800 [4]: 1500 [5]: 3000	Minimum speed for gear stage change
35130[1...5]	GEAR_STEP_MAX_VELO_LIMIT	rpm	[1]: 500 [2]: 1000 [3]: 2000 [4]: 4000 [5]: 8000	Maximum speed of gear stage
35140[1...5]	GEAR_STEP_MIN_VELO_LIMIT	rpm	[1]: 5 [2]: 10 [3]: 20 [4]: 40 [5]: 80	Minimum speed of gear stage
36200[1...5]	AX_VELO_LIMIT	rpm	31944	Threshold value for velocity monitoring

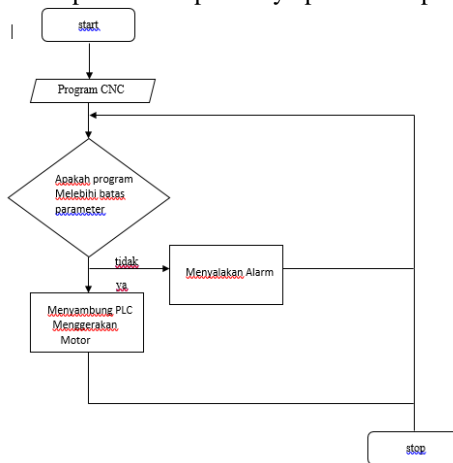
3.4. Simulasi Program

Tahap terakhir dalam perancangan adalah simulasi program. Simulasi ini digunakan untuk mencoba program XML yang sudah dirangkai untuk dicoba pada mesin. Simulasi ini menggunakan Sinumerik 808D on PC yang memiliki tampilan yang sama persis dengan yang ada di mesin CNC tapi yang bisa digunakan hanya tampilan saja.



Gambar 3.5 Simulasi Program

Pada Simulasi program sudah bisa dijalankan seperti pada gambar 3.5. Menunjukkan bahwa program sudah bisa masuk. Tetapi belum bisa membaca data satupun karena tidak ada hardware yang terpasang membuat alarm yang ada diatas gambar bertulisan merah menyala. Untuk alarm biasanya tidak dihiraukan pada simulasi karena memiliki efek apapun. Tetapi jika ada alarm yang menyala pada saat mesin CNC berjalan banyak hal yang perlu diwaspadai karena bisa merusak mesin dan komponen-komponennya percobaan pada mesin.



Gambar 3.6 Flow Chart Program

-----Halaman ini sengaja dikosongkan-----

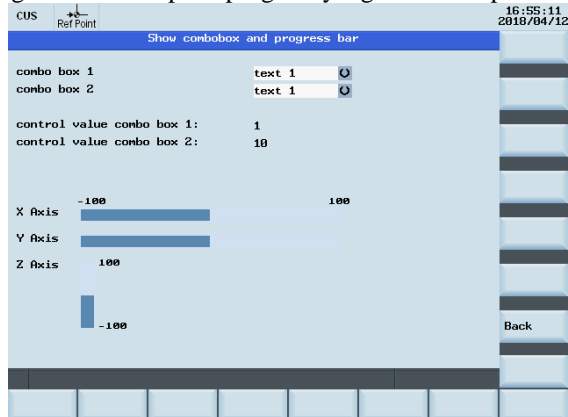
BAB IV

PENGUJIAN DAN ANALISA

Untuk mengetahui apakah tujuan dari pembuatan sistem ini telah tercapai atau tidak, maka perlu diadakannya sebuah pengujian dan analisa terhadap alat yang telah dibuat. Dan sebagai acuan yang tidak terpisahkan adalah adanya proses evaluasi sehingga akan dapat dilakukan langkah-langkah positif guna membawa alat ini kearah yang lebih baik. Adapun pengujian yang dilakukan dengan Dua Tahap.

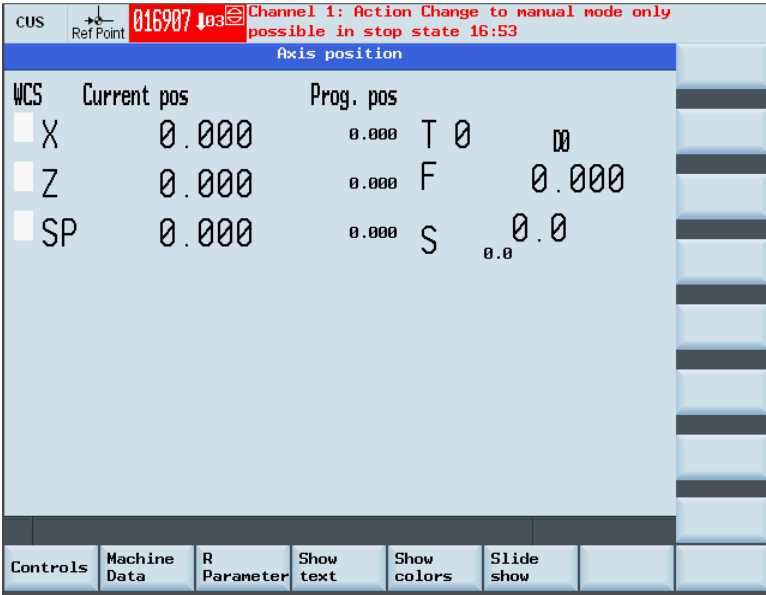
4.1 Pengujian Simulasi Program

Simulasi yang digunakan pada pengujian ini digunakan untuk mencoba program yang di buat dari pemrograman XML untuk menampilkan data pada HMI. Dengan menggunakan simulasi yang ada di PC diharapkan bisa mencegah program yang dapat merusak alat pada mesin CNC untuk itu penggunaan Simulasi sangat diperlukan pada saat pengambilan data pada program yang bisa ditampilkan



Gambar 4.1 Hasil Simulasi

Pengujian Program ini dilakukan di tiap tiap motor yang ada pada mesin CNC. Karena ada 3 motor yang akan diuji maka ada tiga halaman yang diuji pada Simulasi Program. Data yang diambil pada pengujian sensor ini munculnya alarm pada bagian atas tampilan HMI.



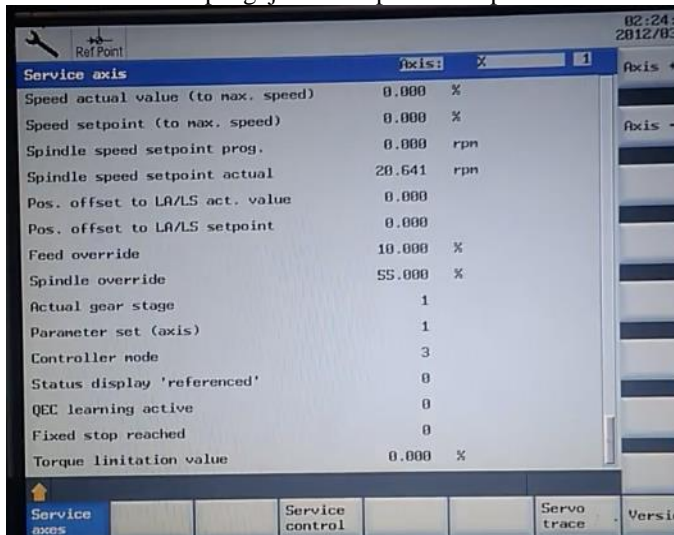
Gambar 4.2 Program Alarm

Pada simulasi program alarm dapat dilihat pada gambar 4.2 hasil dari program simulasi untuk pembuatan program alarm menunjukkan dengan mencoba memasukan program yang melebihi dari parameter yang sudah ditentukan akan muncul alarm pada program diatas untuk memberi tahu pengguna tentang kesalahan pada program yang dimasukan pada mesin Mini CNC untuk itu dapat diketahui program XML yang dibuat sudah bisa dicoba untuk mesin CNC yang bisa digunakan sebagai pengujian

4.2 Pengujian Kecepatan Motor Pada Mini CNC

Pengujian kecepatan motor ini menggunakan mesin Mini CNC. Dengan mengimplementasikan program langsung ke dalam mesin mini CNC. Pengujian ini dilakukan dengan cara memasukan program motor

dengan Kecepatan tertentu. Program yang kecepatan motor yang dimasukan ada 3 yaitu kecepatan 20, 400, 4000 Rpm. Untuk pengambilan data dilihat di halaman pada mesin Mini CNC pada halaman Machine Data. ilustrasi pengujian ini dapat dilihat pada Gambar 4.4 Hasil dari pengujian ini dapat dilihat pada Tabel 4.5.



Service axis	Value	Unit
Speed actual value (to max. speed)	0.000	%
Speed setpoint (to max. speed)	0.000	%
Spindle speed setpoint prog.	0.000	rpm
Spindle speed setpoint actual	20.641	rpm
Pos. offset to LA/LS act. value	0.000	
Pos. offset to LA/LS setpoint	0.000	
Feed override	10.000	%
Spindle override	55.000	%
Actual gear stage	1	
Parameter set (axis)	1	
Controller mode	3	
Status display 'referenced'	0	
QEC learning active	0	
Fixed stop reached	0	
Torque limitation value	0.000	%

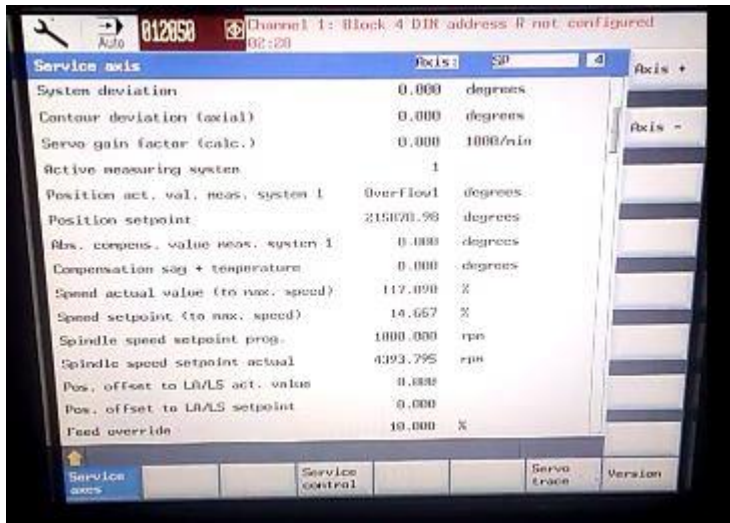
Gambar 4.3 Kecepatan Spindle saat 20 RPM

Tabel 4.1 Hasil Pengujian Kecepatan Spindle

No	Kecepatan Motor Rpm	Status
1.	20	Program Tetap Berjalan
2.	400	Program Tetap Berjalan
3.	4000	Alarm Mesin Nyala

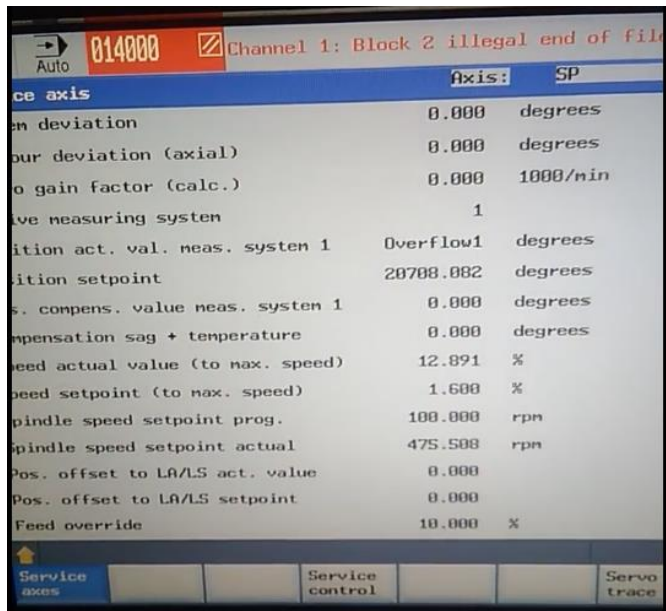
Dari data hasil pengujian tersebut didapatkan dari ketepatan dan Kecepatan Motor dengan pengujian yang dilakukan 3 kali tiap datanya dan program masih bisa digunakan. Pada tabel diatas menunjukan bahwa. Saat motor berputar dengan kecepatan tertentu dengan program yang ditentukan akan menunjukan parameter parameter yang ada untuk menunjukan adanya perubahan pada motor. Pada saat Motor berputar pada 20 Rpm program berjalan seperti biasa dan tidak menunjukan

alarm yang menyala menunjukkan program masih bisa berjalan dan tidak mengganggu pada jalannya program dan tidak memberhentikan motor.



Gambar 4.4 Kecepatan Spindle 4000 Rpm

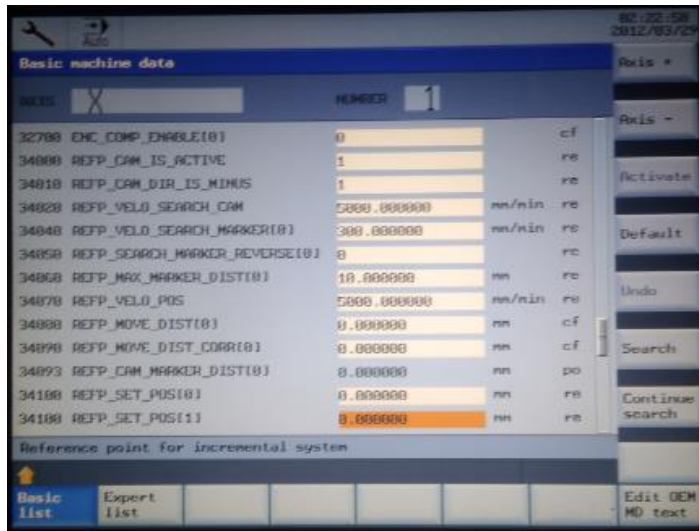
Pada tabel 4.2 dijelaskan bahwa saat motor berputar dengan pelan seperti 400 Rpm program memang menunjukan bahwa ada alarm tetapi alarm disebabkan karena malfungsi hardware lain. Pada kecepatan 400 Rpm motor berputar dengan cepat tetapi tidak melebihi batas parameter yang ditentukan pada mesin CNC. Sedangkan saat motor mesin berputar lebih dari batas maksimal kecepatan pada motor servo seperti 4000 Rpm program langsung menunjukan alarm pada mesin untuk memberi tahu pada pengguna bahwa mesin sedang bermasalah seperti pada gambar 4.5.



Gambar 4.5 Kecepatan Spindle saat 400 RPM

Dengan grafik dan gambar menunjukkan bahwa dengan penambahan program pada mesin bisa digunakan untuk memberi tahu pengguna bahwa mesin sedang bermasalah. Program Parameter memberi pengarahan pada pengguna untuk mengetahui program yang melebihi batas program. Untuk penggunaan dapat menjadi efektif akan diberikan panduan parameter untuk motor sehingga saat batas mesin. Untuk parameter bisa dimasukan oleh perusahaan dan ditentukan untuk memberi batas pada program yang ada. Dari pengujian gerakan Motor berdasarkan perintah Program dapat disimpulkan bahwa gerakan sudah sesuai dengan yang diharapkan yaitu HMI sinumerik sudah bisa digunakan untuk membaca data yang ada

Terdapat juga kendala yaitu saat kecepatan 400 Rpm masih ada program alarm untuk hardware yang lain karena sedang adanya perbaikan.



Gambar 4.6 Program Parameter

Nilai daripada data yang terbaca sangat bervariasi, mulai dari nilai positif. Hal ini diakibatkan oleh noise yang sangat besar yang terkandung didalam sensor. Nilai yang bervariasi ini dapat ditemukan pada data.

Dari pengujian seperti pada gambar 4.5 mengambil data berupa kecepatan motor mengikuti program CNC pada axis yang ada. Disini data yang dibandingkan adalah data pada kecepatan motor dengan perbedaan kecepatan untuk mencoba agar apakah alarm bisa menyala dan menghentikan motor.

BAB V

PENUTUP

Bab ini berisi tentang kesimpulan-kesimpulan yang didapatkan selama proses pembuatan Tugas Akhir ini beserta saran-saran untuk perbaikan dan pengembangannya.

5.1 Kesimpulan

Bedasarkan Pengujian yang dilakukan pada HMI ini bisa ditarik kesimpulan bahwa alat ini sudah bisa digunakan untuk mengumpulkan data pada mesin Mini CNC untuk pengawasan komponen. Dari yang bisa diambil datanya adalah motor Spindle untuk kecepatan dan jarak yang ditempuh pada motor bisa dilihat dengan menggunakan program pada HMI yang di setting dengan menggunakan Program XML

Proyek Tugas Akhir ini merupakan Tugas Akhir yang dilaksanakan untuk menyelesaikan masalah pada perusahaan PT. CNC Controller Indonesia

5.2 Saran

Saran –saran yang dapat diberikan untuk implementasi dan pengembangan lebih lanjut dari sistem ini Dalam proses pembacaan dapat dikembangkan tidak hanya motor *spindle* saja .

-----Halaman ini sengaja dikosongkan-----

DAFTAR PUSTAKA

- [1] F. Wu, "CNC milling machine spindle characteristics analysis by FEM," *2010 Int. Conf. Intell. Comput. Technol. Autom. ICICTA 2010*, vol. 1, pp. 954–957, 2010.
- [2] G. S. Dalmasius, *Pemrograman CNC dan Aplikasi Didunia Industri*. 2010.
- [3] D. Build, *Introduction to CNC Machining NC Lathe* $\equiv T$, vol. 39, no. 37. .
- [4] O. Thomas and M. Fellmann, *Extensible Markup Language*, vol. 37, no. 7. 2008.
- [5] M. S. Li and F. F. Zhang, "The research of embedded CNC system component," *ICCASM 2010 - 2010 Int. Conf. Comput. Appl. Syst. Model. Proc.*, vol. 7, no. Iccasm, pp. 601–604, 2010.
- [6] L. Qiu, "Preventive Maintenance Schedule of CNC Machine Tool Based on Monte Carlo Simulation," *2014 17th Int. Conf. Electr. Mach. Syst. ICEMS 2014*, pp. 993–996, 2015.
- [7] O. Instructions, "SINUMERIK SINUMERIK 808D ADVANCED Commissioning Manual," pp. 1–172, 2013.
- [8] M. E. Brumbach and J. A. Clade, *Industrial Maintenance*. 2003.
- [9] R. Benz, O. Fröhlich, P. Läger, and M. Montal, *Electrical capacity of black lipid films and of lipid bilayers made from monolayers*, vol. 394, no. 3. 1975.
- [10] CIBSE, *Maintenance engineering and management: Guide M*. 2008.
- [11] S. Krar, "C OMPUTER N UMERICAL Championships," *Textbook*, p. 40, 1990.

-----Halaman ini sengaja dikosongkan-----

LAMPIRAN

A. Listing Program pada XML

```
<form name="Menu1">
  <paint>
    <text xpos="213" color="#000000" ypos="39"
font="2">CONDITION AXIS X</text>
    <text xpos="38" color="#000000" ypos="199"
font="2">TOTAL DISTANCE</text>
    <text xpos="36" color="#000000" ypos="134"
font="2">DISTANCE</text>
    <text xpos="39" color="#000000" ypos="75"
font="2">SPEED</text>
  </paint>
  <init>
    <caption>User Mask</caption>
    <control name="OutputField1" xpos="42" ypos="236"
height="18" width="80"
refvar="Nck/Channel/Parameter/30240[0]" hotlink="true"
fieldtype="readonly" format="% 10.3f" time="super fast"
font="1" color_bk="#ffffff" />
    <control name="OutputField2" xpos="42" ypos="168"
height="18" width="80"
refvar="Nck/Channel/Parameter/32260[0]" hotlink="true"
fieldtype="readonly" format="% 10.3f" time="super fast"
font="1" color_bk="#ffffff" />
    <control name="OutputField3" xpos="46" ypos="108"
height="18" width="80"
refvar="Nck/Channel/Parameter/34200[0]" hotlink="true"
fieldtype="readonly" format="% 10.3f" time="super fast"
font="1" color_bk="#ffffff" />
    <control name="OutputField1" xpos="42" ypos="236"
height="18" width="80"
refvar="Nck/Channel/Parameter/32100[2048]" hotlink="true"
fieldtype="readonly" format="% 10.3f" time="super fast"
font="1" color_bk="#ffffff" />
    <control name="OutputField1" xpos="42" ypos="236"
height="18" width="80"
refvar="Nck/Channel/Parameter/30202[0]" hotlink="true">
```

```

fieldtype="readonly" format="% 10.3f" time="super fast"
font="1" color_bk="#ffffff" />
<control name="OutputField1" xpos="42" ypos="236"
height="18" width="80"
refvar="Nck/Channel/Parameter/30240[2]" hotlink="true"
fieldtype="readonly" format="% 10.3f" time="super fast"
font="1" color_bk="#ffffff" />
<control name="OutputField1" xpos="42" ypos="236"
height="18" width="80"
refvar="Nck/Channel/Parameter/30130[0]" hotlink="true"
fieldtype="readonly" format="% 10.3f" time="super fast"
font="1" color_bk="#ffffff" />
<control name="OutputField1" xpos="42" ypos="236"
height="18" width="80"
refvar="Nck/Channel/Parameter/30120[1]" hotlink="true"
fieldtype="readonly" format="% 10.3f" time="super fast"
font="1" color_bk="#ffffff" />
<control name="OutputField1" xpos="42" ypos="236"
height="18" width="80"
refvar="Nck/Channel/Parameter/35100[0]" hotlink="true"
fieldtype="readonly" format="% 10.3f" time="super fast"
font="1" color_bk="#ffffff" />
<control name="OutputField1" xpos="42" ypos="236"
height="18" width="80"
refvar="Nck/Channel/Parameter/30160[3]" hotlink="true"
fieldtype="readonly" format="% 10.3f" time="super fast"
font="1" color_bk="#ffffff" />
<control name="OutputField1" xpos="42" ypos="236"
height="18" width="80"
refvar="Nck/Channel/Parameter/35140[4]" hotlink="true"
fieldtype="readonly" format="% 10.3f" time="super fast"
font="1" color_bk="#ffffff" />
<control name="OutputField1" xpos="42" ypos="236"
height="18" width="80"
refvar="Nck/Channel/Parameter/34201[3000]" hotlink="true"
fieldtype="readonly" format="% 10.3f" time="super fast"
font="1" color_bk="#ffffff" />
<control name="OutputField1" xpos="42" ypos="236"
height="18" width="80"

```



```

refvar="Nck/Channel/Parameter/32500[0]" hotlink="true"
fieldtype="readonly" format="%10.3f" time="super fast"
font="1" color_bk="#ffffff" />
<control name="OutputField1" xpos="42" ypos="236"
height="18" width="80"
refvar="Nck/Channel/Parameter/31440[0]" hotlink="true"
fieldtype="readonly" format="%10.3f" time="super fast"
font="1" color_bk="#ffffff" />
<control name="OutputField1" xpos="42" ypos="236"
height="18" width="80"
refvar="Nck/Channel/Parameter/30240[0]" hotlink="true"
fieldtype="readonly" format="%10.3f" time="super fast"
font="1" color_bk="#ffffff" />
<form name="Menu1">
  <paint>
    <text xpos="213" color="#000000" ypos="39"
font="2">CONDITION AXIS X</text>
    <text xpos="38" color="#000000" ypos="199"
font="2">TOTAL DISTANCE</text>
    <text xpos="36" color="#000000" ypos="134"
font="2">DISTANCE</text>
    <text xpos="39" color="#000000" ypos="75"
font="2">SPEED</text>
  </paint>
  <init>
    <caption>User Mask</caption>
    <control name="OutputField1" xpos="42" ypos="236"
height="18" width="80"
refvar="Nck/Channel/Parameter/30240[0]" hotlink="true"
fieldtype="readonly" format="%10.3f" time="super fast"
font="1" color_bk="#ffffff" />
    <control name="OutputField2" xpos="42" ypos="168"
height="18" width="80"
refvar="Nck/Channel/Parameter/32260[0]" hotlink="true"
fieldtype="readonly" format="%10.3f" time="super fast"
font="1" color_bk="#ffffff" />
    <control name="OutputField3" xpos="46" ypos="108"
height="18" width="80"
refvar="Nck/Channel/Parameter/34200[0]" hotlink="true"

```

```

fieldtype="readonly" format="%10.3f" time="super fast"
font="1" color_bk="#ffffff" />
<control name="OutputField1" xpos="42" ypos="236"
height="18" width="80"
refvar="Nck/Channel/Parameter/32100[2048]" hotlink="true"
fieldtype="readonly" format="%10.3f" time="super fast"
font="1" color_bk="#ffffff" />
<control name="OutputField1" xpos="42" ypos="236"
height="18" width="80"
refvar="Nck/Channel/Parameter/30202[0]" hotlink="true"
fieldtype="readonly" format="%10.3f" time="super fast"
font="1" color_bk="#ffffff" />
<control name="OutputField1" xpos="42" ypos="236"
height="18" width="80"
refvar="Nck/Channel/Parameter/30240[2]" hotlink="true"
fieldtype="readonly" format="%10.3f" time="super fast"
font="1" color_bk="#ffffff" />
<control name="OutputField1" xpos="42" ypos="236"
height="18" width="80"
refvar="Nck/Channel/Parameter/30130[0]" hotlink="true"
fieldtype="readonly" format="%10.3f" time="super fast"
font="1" color_bk="#ffffff" />
<control name="OutputField1" xpos="42" ypos="236"
height="18" width="80"
refvar="Nck/Channel/Parameter/30120[1]" hotlink="true"
fieldtype="readonly" format="%10.3f" time="super fast"
font="1" color_bk="#ffffff" />
<control name="OutputField1" xpos="42" ypos="236"
height="18" width="80"
refvar="Nck/Channel/Parameter/35100[0]" hotlink="true"
fieldtype="readonly" format="%10.3f" time="super fast"
font="1" color_bk="#ffffff" />
<control name="OutputField1" xpos="42" ypos="236"
height="18" width="80"
refvar="Nck/Channel/Parameter/30160[3]" hotlink="true"
fieldtype="readonly" format="%10.3f" time="super fast"
font="1" color_bk="#ffffff" />
<control name="OutputField1" xpos="42" ypos="236"
height="18" width="80"

```

```

refvar="Nck/Channel/Parameter/35140[4]" hotlink="true"
fieldtype="readonly" format="%10.3f" time="super fast"
font="1" color_bk="#ffffff" />
<control name="OutputField1" xpos="42" ypos="236"
height="18" width="80"
refvar="Nck/Channel/Parameter/34201[3000]" hotlink="true"
fieldtype="readonly" format="%10.3f" time="super fast"
font="1" color_bk="#ffffff" />
<control name="OutputField1" xpos="42" ypos="236"
height="18" width="80"
refvar="Nck/Channel/Parameter/32500[0]" hotlink="true"
fieldtype="readonly" format="%10.3f" time="super fast"
font="1" color_bk="#ffffff" />
<control name="OutputField1" xpos="42" ypos="236"
height="18" width="80"
refvar="Nck/Channel/Parameter/31440[0]" hotlink="true"
fieldtype="readonly" format="%10.3f" time="super fast"
font="1" color_bk="#ffffff" />
<control name="OutputField1" xpos="42" ypos="236"
height="18" width="80"
refvar="Nck/Channel/Parameter/30240[0]" hotlink="true"
fieldtype="readonly" format="%10.3f" time="super fast"
font="1" color_bk="#ffffff" />
}

```

-----Halaman ini sengaja dikosongkan-----

LAMPIRAN B

SINUMERIK 808D and SINUMERIK 808D ADVANCED

Optimal scalability in the entry class



Scalable CNC performance

SINUMERIK 808D and SINUMERIK 808D ADVANCED offer a broad performance range for entry class machines. With the drive bus technology, SINUMERIK 808D ADVANCED together with SINAMICS V70 and SIMOTICS S-1FL6 guarantees a high system performance. Both, SINUMERIK 808D and SINUMERIK 808D ADVANCED, offer a high cost-performance ratio for basic standard machines.

SINAMICS and SIMOTICS

The powerhouses behind the scenes

	 <ul style="list-style-type: none"> Rated torque 1.9–40 Nm Drive line input 400V Up to 300% overload 2 encoder options: Incremental encoder with 2,500 ppr; Absolute encoder with 20-bit resolution Motor speed up to 4000 rpm Safe Torque Off (STO) Degree of protection IP65 	SINAMICS V70 SIMOTICS S-1FL6
	 <ul style="list-style-type: none"> Rated torque 4–10 Nm Drive line input 220V 200% overload Incremental encoder with 2500 ppr Motor speed 2000 rpm Degree of protection IP54 	SINAMICS V60 SIMOTICS S-1FL5

SINUMERIK 808D / 808D ADVANCED, in conjunction with SINAMICS drives and SIMOTICS motors, are optimally designed to address the requirements of basic standard turning and milling machines.

	SINUMERIK 808D	SINUMERIK 808D ADVANCED
	PLU141.1	PLU16a.2
Configuration		
Mechanical design	Panel-based	
Operation with SINAMICS V60 and SIMOTICS S-1FL5 via pulse / direction interface		—
Operation with SINAMICS V70 and SIMOTICS S-1FL6 via drive bus interface	—	
Maximum number of axes / spindles	4	5
a 10V analog interface for spindles		
CNC user memory	1.25 MB	
Additional CNC user memory on USB stick		
Display size (TFT color displays)	7.5"	
PLC adaptation control	S7-200-based	
Standard data transfer RS232C		
Standard data transfer via USB		
Standard data transfer via Ethernet	—	
Number of digital I/O	72 / 48	72 / 48
Tool probe interface		
Axis functions		
Acceleration with jerk limitation		
Dynamic Servo Control in the drive	—	
Feed motor, max. speed	2000 rpm	Up to 4000 rpm
Drive and motor overload capability	200%	Up to 200%
2500 ppr incremental encoders are supported		
20-bit absolute encoders are supported	—	
Interpolation		
Interpolating axes, up to (turning / milling)	3 / 4	3 / 4
Straight line, circle, helix		
Thread cutting with constant or variable pitch		
Rigid tapping		
Advanced Surface	Milling	
Look Ahead, number of blocks (turning / milling)	1 / 50	1 / 50
Compressor	—	
Couplings		
Synchronous axis pair (gantry basic)	—	
Transformations		
Face / peripheral surface transformation (TRANSMIT / TRACYL without Y axis)	—	

	SINUMERIK 800D	SINUMERIK 800D ADVANCED
	PPU141.1	PPU164.2
SINUMERIK synchronous architecture		
Asynchronous subprograms ASUB		
Compensations		
Backlash and lead screw compensation		
Bidirectional lead screw compensation	—	
Friction compensation	—	
Tools		
Number of tool/cutting edges in the tool list, up to	64 / 128	64 / 128
Driven tool for turning	—	
CNC operation		
SINUMERIK Operate BASIC		
T, S, M function in JOG		
Graphic guided tool / workplace measuring in JOG		
Block search with calculation (T, S, F, M, position)		
SINUMERIK 800D on PC (can be downloaded free of charge)		
CNC programming		
SINUMERIK CNC programming language with high level language elements		
ISO code CNC programming language with canned cycles		
ProgramGUIDE BASIC (technology cycle support)		
Technology cycles for drilling, milling and turning		
Contour computer		
2D CNC simulation and simultaneous recording (real-time simulation)		
Onboard optimization and diagnostics		
Context-sensitive onboard help system		
Onboard signal and network diagnostics		
Onboard servo and drive optimization (AST)	—	
Onboard drive commissioning and diagnosis	—	
Service planner		
Safety functions		
Safe Torque Off (drive-based)	—	
Open Architecture		
Customized HMI extension (Easy X ML)		
User cycle		
— not available available (certain functions are available as CNC option, please ask your machine tool manufacturer)		

Biodata Penulis



Penulis bernama Arinto Eka Prasetyo lahir di Surabaya, provinsi Jawa Timur pada tanggal 18 April 1998 dari pasangan Sugianto dan Endang. Penulis merupakan anak pertama dari dua bersaudara. Penulis menyelesaikan sekolah di SDN Rangkah VIII, melanjutkan di Sekolah Menengah Pertama Negeri 6 Surabaya dan menyelesaikan Sekolah Menengah Atas YP Trisila Surabaya. melanjutkan di Institut Teknologi Sepuluh

Nopember, Departemen Teknik Elektro Otomasi, Surabaya. Penulis merupakan anggota laboratorium Automation dan Computer System selama 2 tahun. Penulis juga pernah menjadi panitia acara Industrial Automation and Robotic Competition 2 tahun dan Menjadi panitia acara Siemens Science & Technology Competition.

Penulis dapat dihubungi melalui email : arintoeka@gmail.com